Яндекс Такси

Что под капотом у платформы данных Яндекс.Такси?

Евгений Ермаков, архитектор DMP

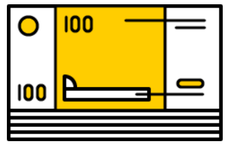
Платформа данных Яндекс.Такси

250

Уникальных пользователей BI-системы в день

900

Отчетов по различным тематикам



3

Крупных бизнес-юнита: Такси, Еда и Лавка

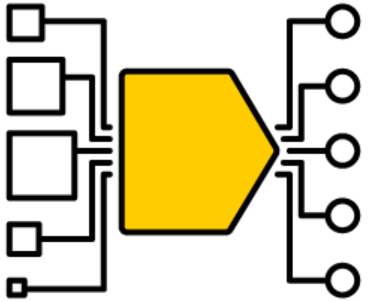
1 Пб

Накопленных данных по трем бизнес-юнитам

Платформа данных Яндекс.Такси

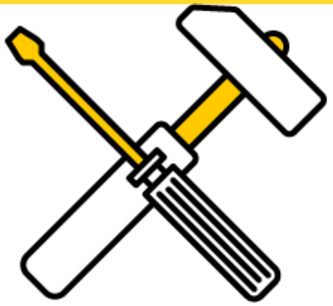
Технологии

Организация



Платформа данных Яндекс.Такси

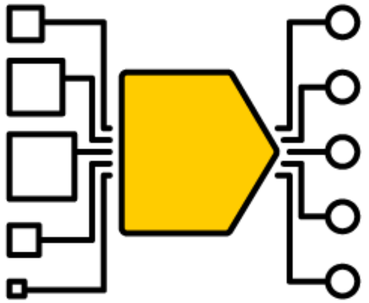
Технологии



Инструменты

Архитектура

Организация

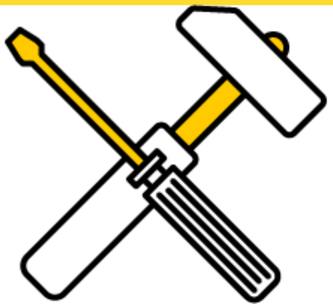


Команда

Процессы

Платформа данных Яндекс.Такси

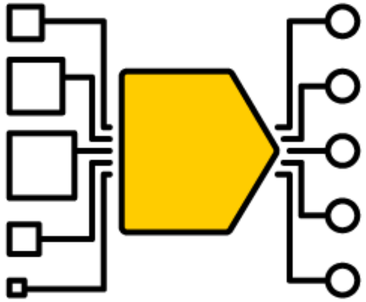
Технологии



Инструменты

Архитектура

Организация



Команда

Процессы

- Какие инструменты обработки и хранения данных используются в Я.Такси?
- Какие Архитектурные принципы заложены в Хранилище?
- Как организована команда?
- Какую роль выполняют инженеры данных?

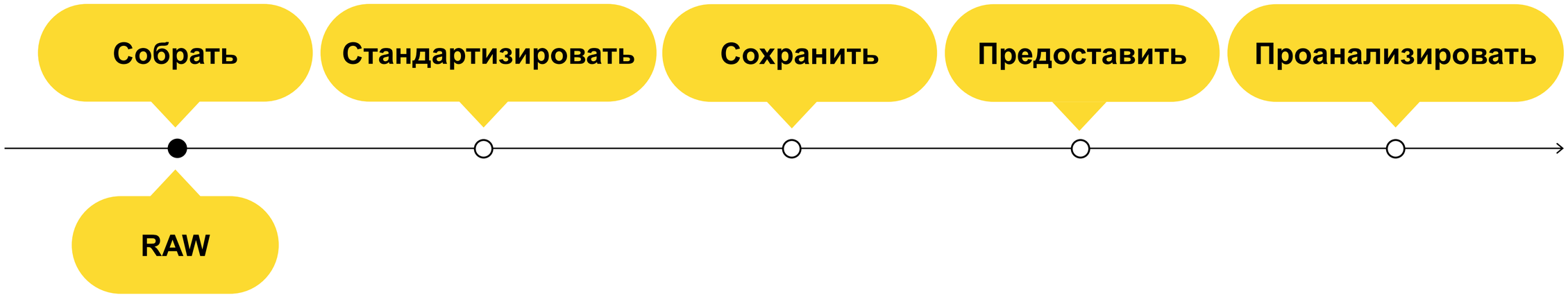
01. Технологии

Инструменты работы с данными и архитектура

- Архитектура слоев
- Инструменты
- Детальный слой
- Гибридная модель

Архитектура слоев данных

Архитектура слоев данных



Цель

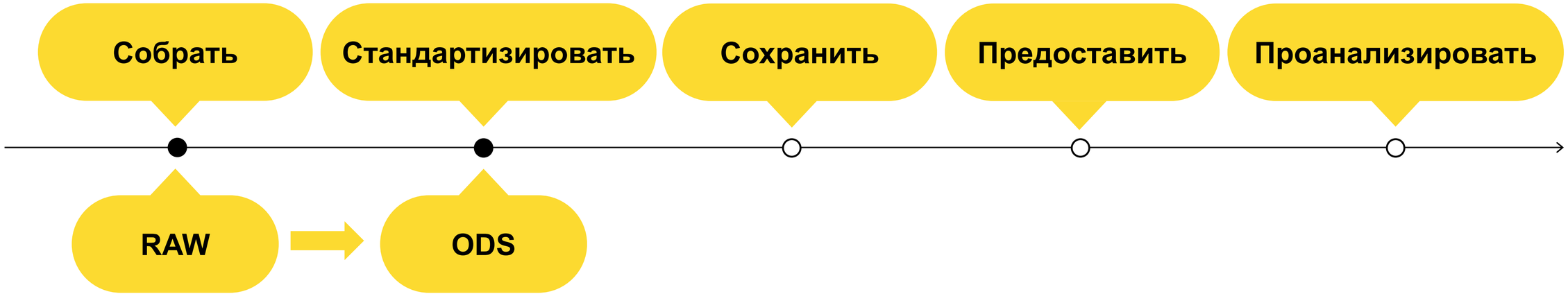
› Захватить сигналы источника

Задачи

› собрать данные с источника **as-is**

› преобразовать их в объекты с понятным описанием и методом доступа

Архитектура слоев данных



Цель

› Хранить операционные данные источника

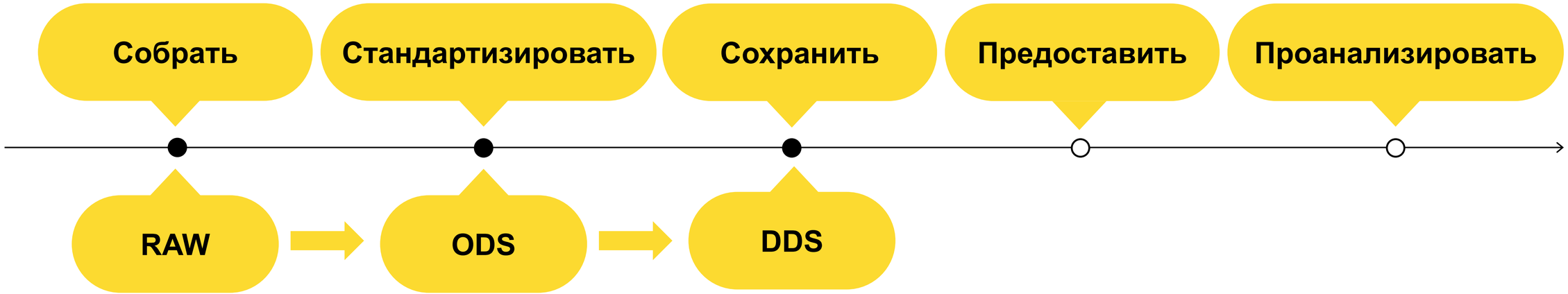
Задачи

› сформировать набор сущностей источника

› разложить данные сущностям

› предоставить стандартный интерфейс доступа к данным

Архитектура слоев данных



Цель

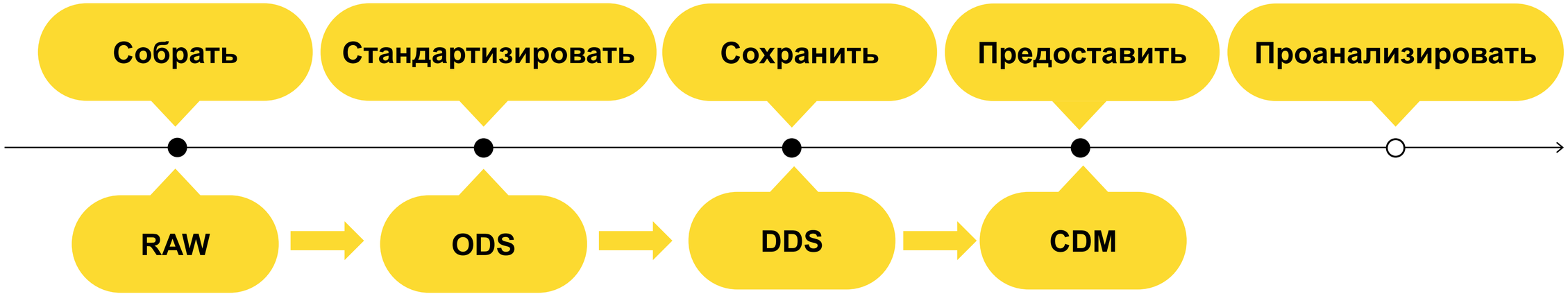
› Накапливать данные о сущностях доменной модели

Задачи

› Хранить детальную историю изменений

› Консолидировать данные между источниками

Архитектура слоев данных



Цель

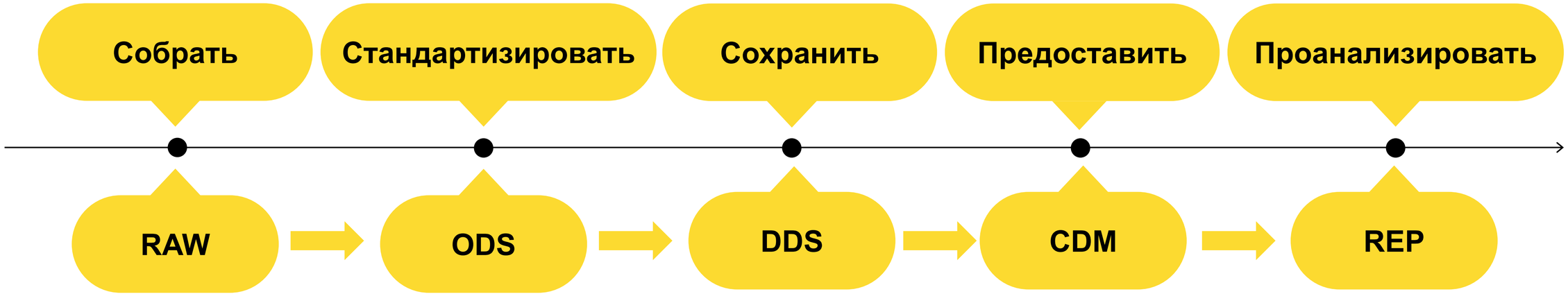
› Предоставлять витрины данных для анализа

Задачи

› Формировать данные в контексте бизнес-потребностей

› Оптимизировать доступ на чтение

Архитектура слоев данных



Цель

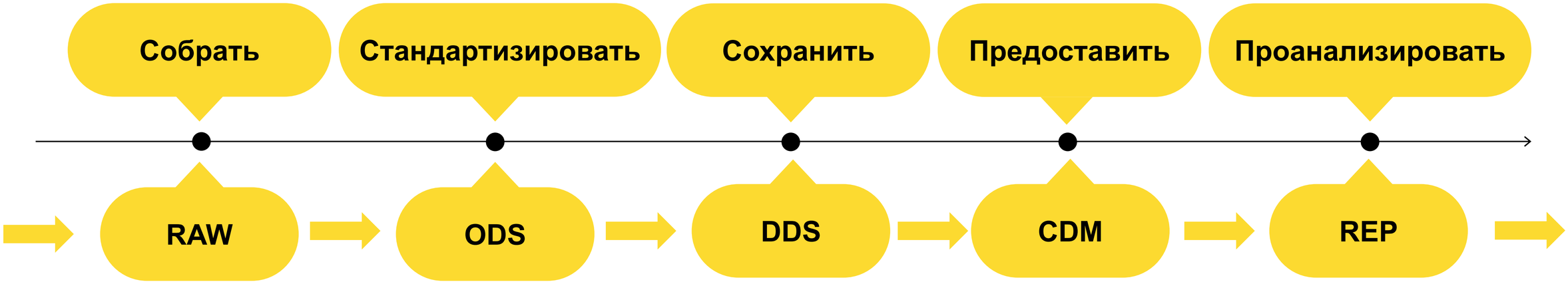
› Хранить отчетные срезы

Задачи

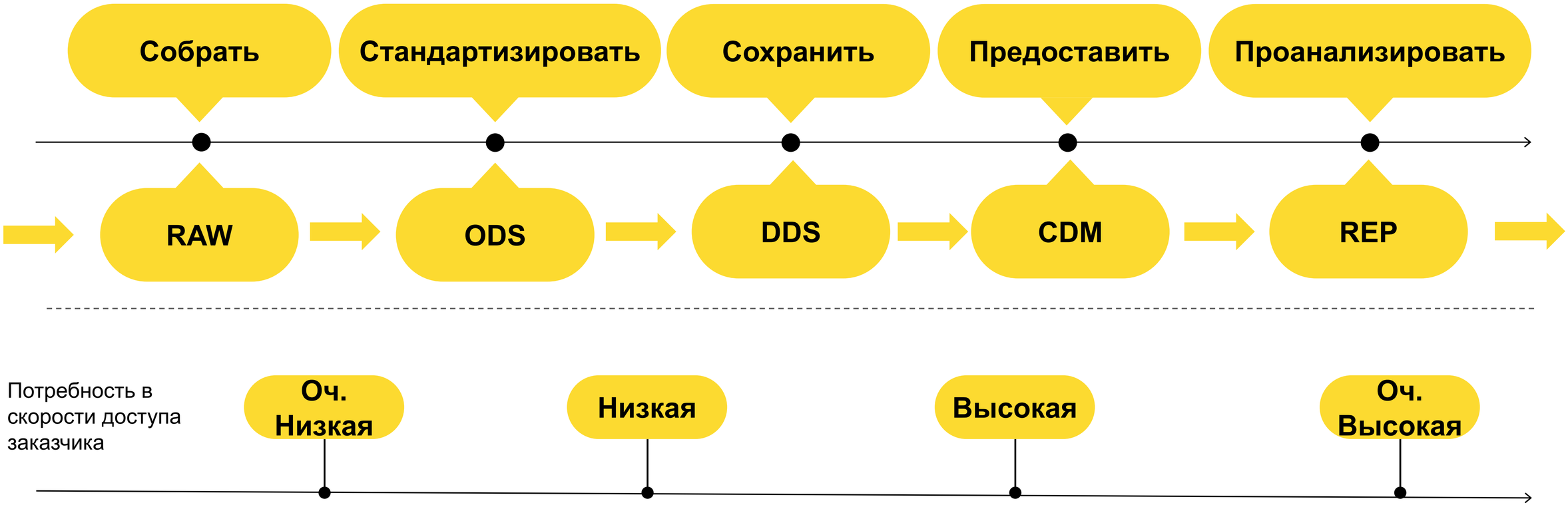
› Формировать данные в контексте бизнес-потребностей

› Готовить агрегированные отчеты

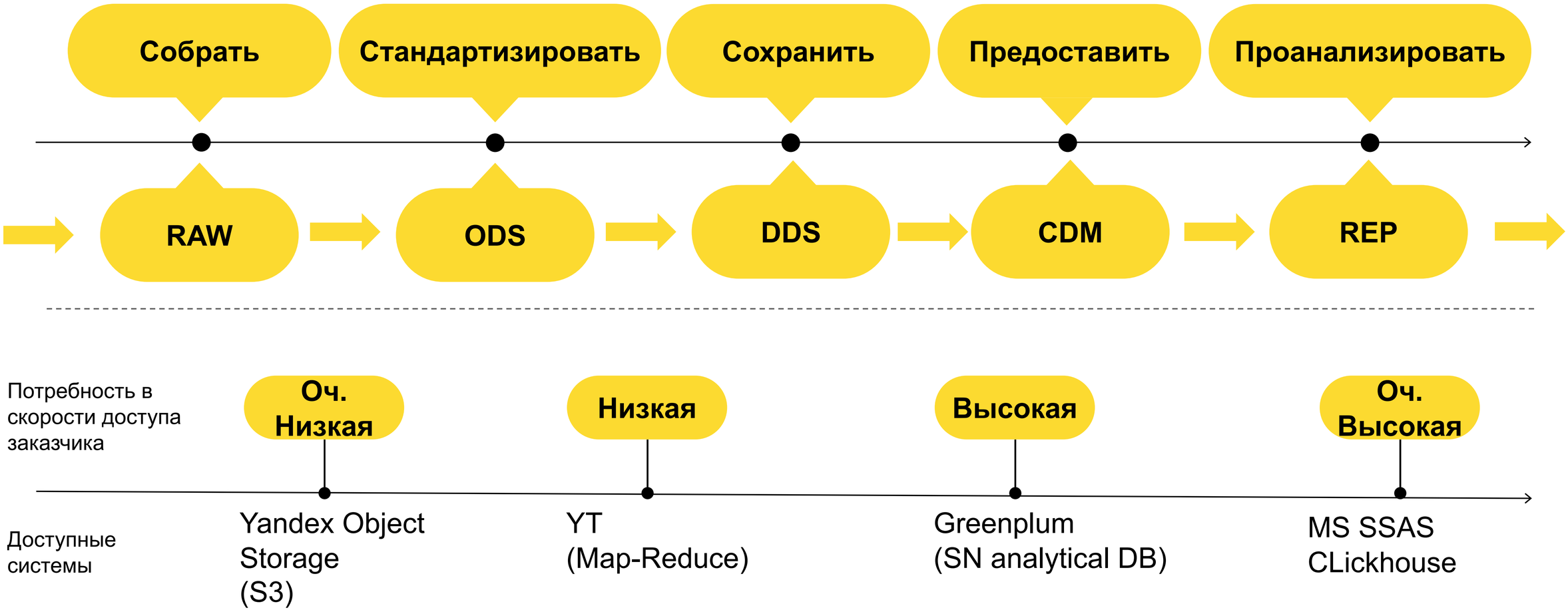
Архитектура слоев данных



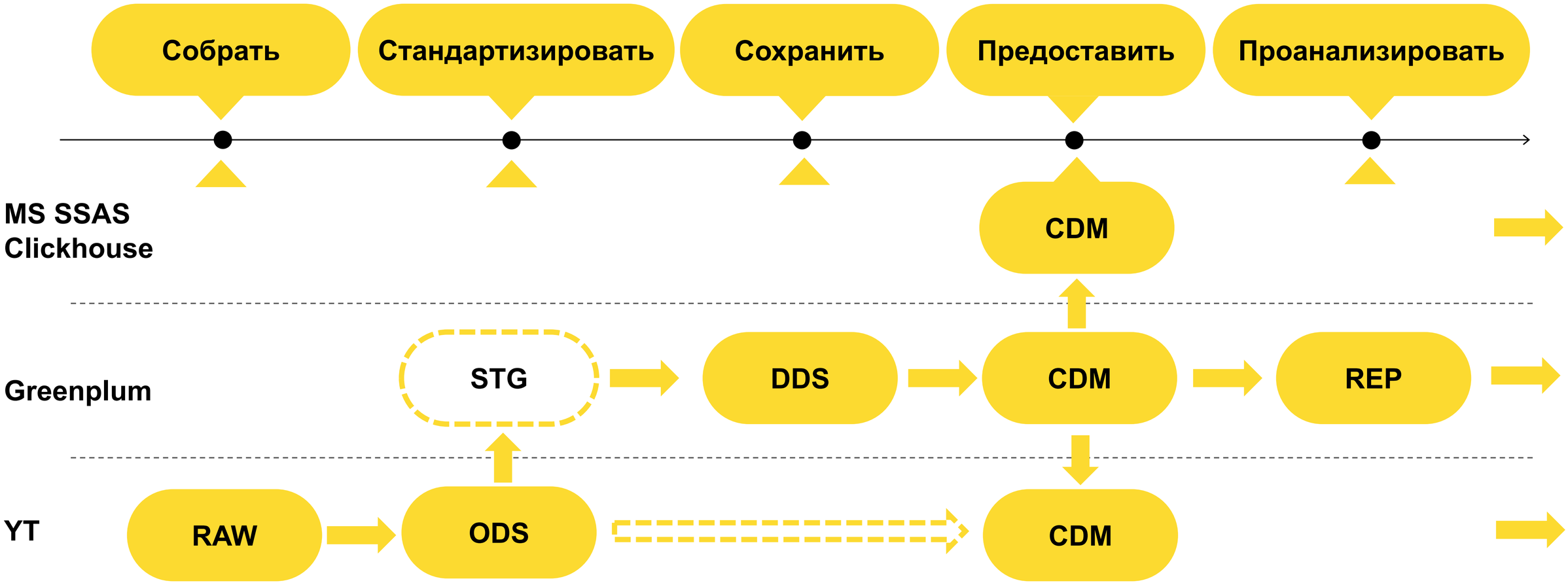
Архитектура слоев данных



Архитектура слоев данных



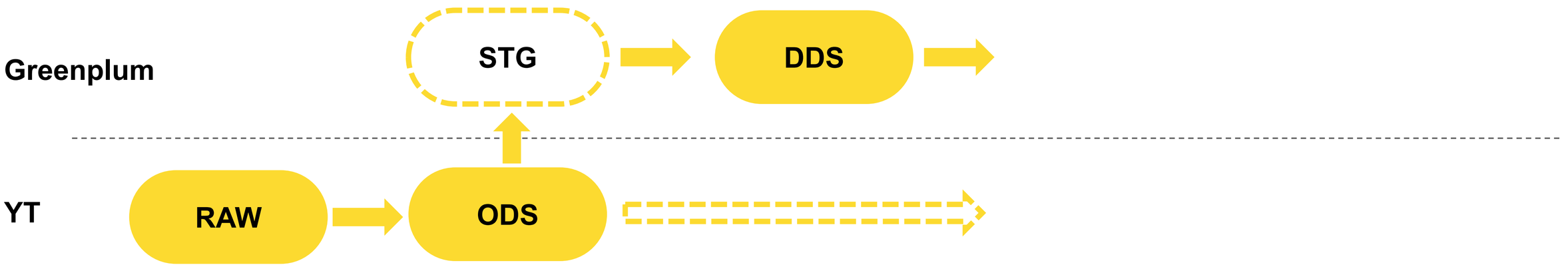
Архитектура слоев данных



Захват данных

Устойчив к изменению данных на источнике

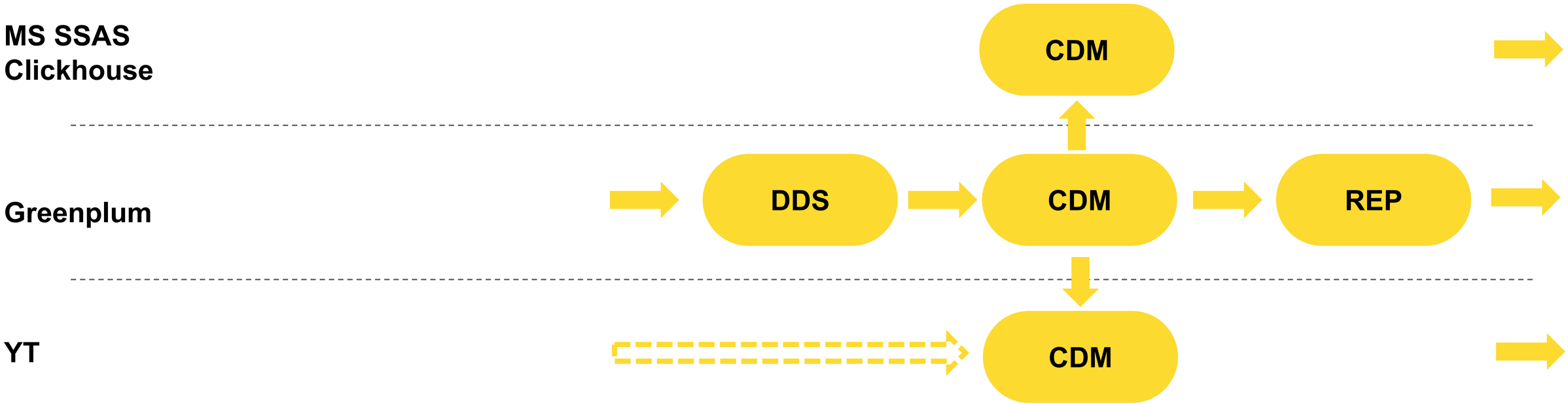
- Инкремент поступает через сервис репликации данных
- Полносрезные данные поступают напрямую в YT
- Формат данных в RAW изолирован от структуры источника
- ODS нормализован и содержит только нужную информацию



Представление данных

Витрины и отчеты доступны в разных источниках

- Строятся по инкременту
- Могут быть историзированы как по бизнес-дате, так и по технической дате
- Денормализованы и оптимизированы под чтение



Детальный слой

Детальный слой – ключевой для построения доменной модели

- Хранить историю изменений сущностей
- Отвечает за консолидацию данных между источниками
- Устойчив к изменению в бизнесе
- Модульный и масштабируемый

Greenplum

Подходы к проектированию

сложность эксплуатации, простота внесения изменений

Никакого	Звезда и снежинка	Data Vault	Anchor modeling
› Денормализация	› Нормализация	› Строгая нормализация	› Ультра нормализация
› Можно использовать без подготовки	› Можно использовать с минимальной подготовкой	› Нельзя использовать без подготовки	› Нельзя использовать без подготовки
		› Не надо перестраивать	› Не надо перестраивать
› Неустойчиво к изменениям	› Неудобно перестраивать		
› Дублирование информации		› Нет дублирования информации	› Нет дублирования информации
	› Минимальное дублирование информации		
› Нет join	› Приемлемое количество join	› Большое количество join	› Ультра количество join

легкость эксплуатации, сложность внесения изменений

Highly Normalized Hybrid Model (hNhM)

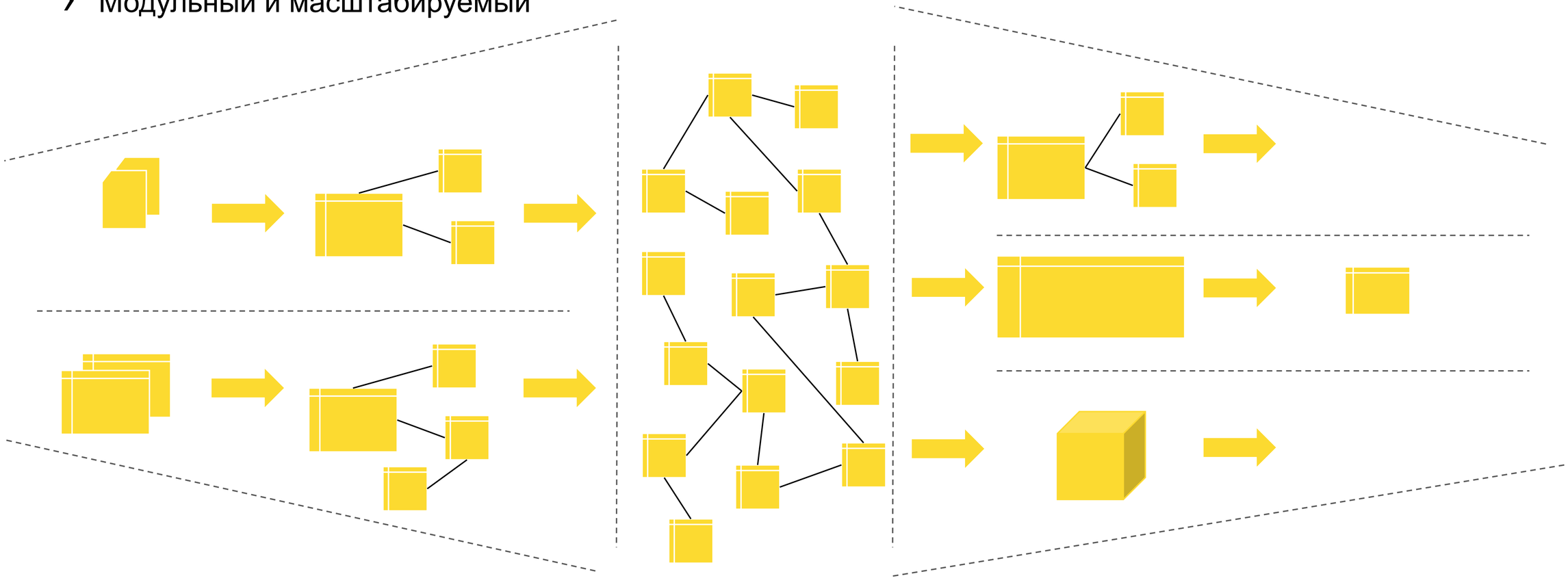
Выбирать оптимальный формат хранения для каждого конкретного случая

- Высокая нормализация
- Параллельная загрузка из разных источников
- Устойчив к изменению в бизнесе
- Идемпотентный к повторной загрузке
- Модульный и масштабируемый

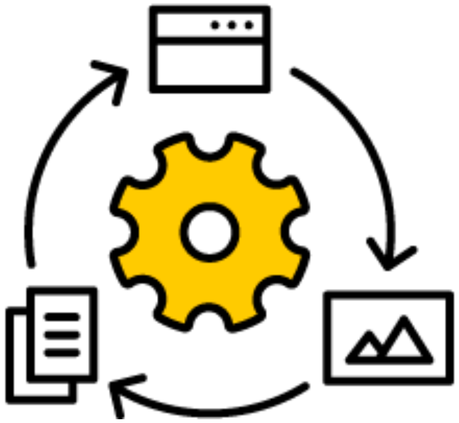
Highly Normalized Hybrid Model (hNhM)

Выбирать оптимальный формат хранения для каждого конкретного случая

- Высокая нормализация
- Параллельная загрузка из разных источников
- Устойчив к изменению в бизнесе
- Идемпотентный к повторной загрузке
- Модульный и масштабируемый



Автоматизация



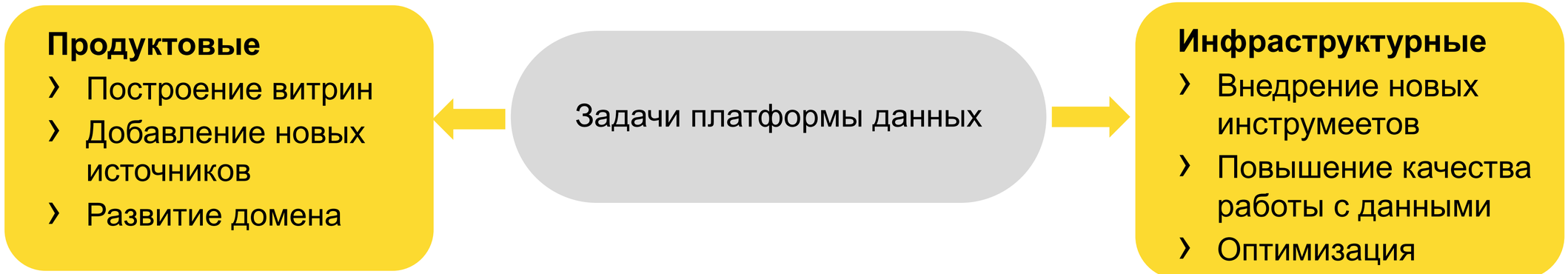
- Захват данных происходит через сервис репликации, гарантирующий доставку
- Изменение структуры данных не влияет на процесс (кроме первичного ключа и поля партицирования)
- Витрины строятся по инкременту, который формируется автоматизированно
- Манипуляции с сущностями в детальном слое стандартизованы
- Заказчик может выбрать удобный для себя интерфейс доступа к данным
- Мониторинг проблем стандартными средствами

02. Организация

Процессы и роли в команде

- Разделение на команды
- Роли и ответственность
- Цели и задачи
- Инженеры данных

Организация процесса



Организация процесса

Организация процесса

Стандартная задача проходит следующие этапы:

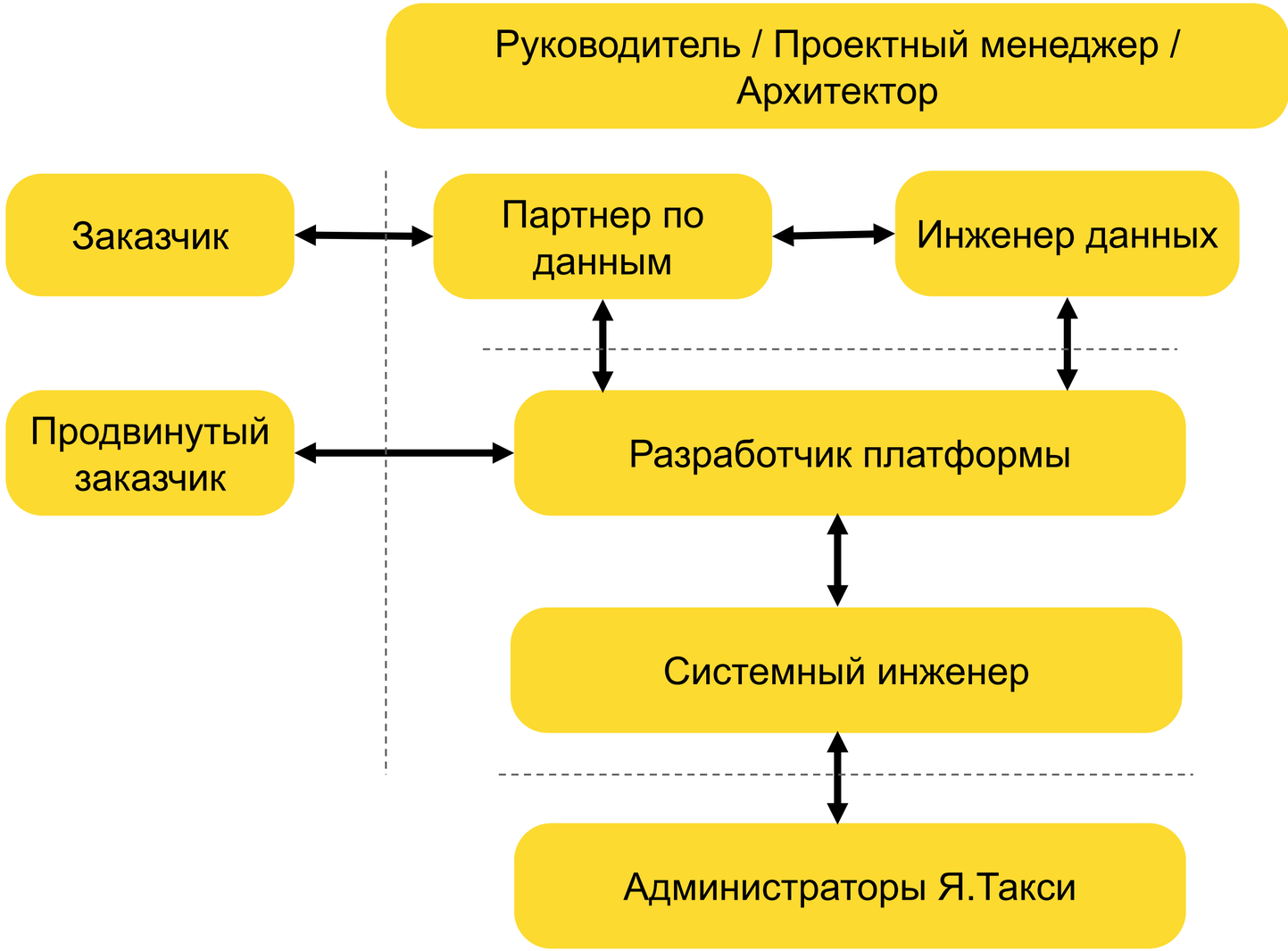
1.Заказчик формирует запрос на новые данные или новую витрину

2.Выделенный партнер по данным структурирует мысль и вносит изменения в модель данных с помощью платформы

3.Инженер данных реализует расчет объекта с помощью платформы

4.Заказчик с парнером по данным проверяют объект

5.Заказчик доволен



Роли

Менеджер проектов

- Курирует взаимодействие между подразделениями
- Ведет крупные проекты

Архитектор

- Отвечает за системность
- Контролирует корректное использование инструментов

Разработчик платформы

- Повышает качество работы ETL-платформы
- Автоматизирует работу

Партнер по данным

- Отвечает за данные как за продукт
- Повышает качество работы с данными

Инженер данных

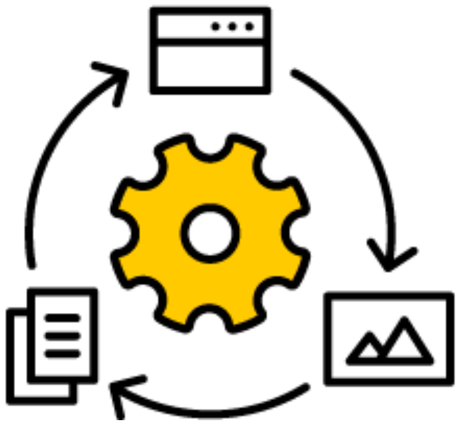
- Разрабатывает сложные ETL-процессы
- Стандартизирует подходы к работе с однотипными источниками

Системный инженер

- Поддерживает сервисы и системы
- Отвечает за интеграцию с внешними системами

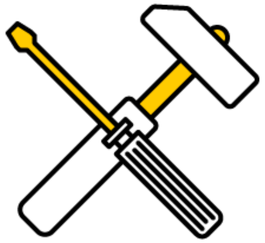
Взаимодействие DP и DE

Автоматизация

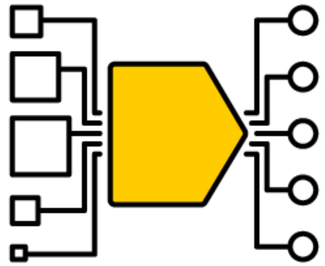


- Вся артефакты работы (в том числе и по анализу данных: метаданные объектов, маппинги) фиксируется в репозитории
- Ревью проходит стандартными способами git (с автоматизацией распределения ревьюеров)
- Документация, data lineage и зависимости строятся из кода
- Все процессы логируются и доступны для последующего анализа в MetaDWH

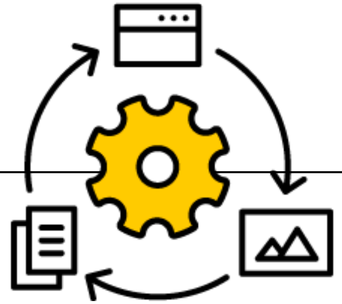
Платформа данных Яндекс.Такси

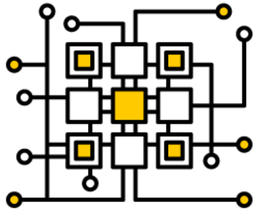


Технологии



Организация





Инструменты

- YT->GP->CH\MS SSAS
- hNhM как модель данных в DDS

Архитектура

- Системность
- Строгость
- Прозрачность

Команда

- Разделение на инфраструктурное и продуктовое направления
- Борьба и единство противоположностей

Процессы

- Отдельные продуктовые команды на области данных
- Автоматизация на всех этапах работы

Яндекс Такси

Спасибо

Евгений Ермаков

архитектор

 jkermakov@yandex-team.ru

 @iJKos